

DIAL

Distributed Interactive Analysis of Large datasets

ATLAS SW Workshop

Grid session

David Adams

BNL

May 15, 2003



Contents

Goals of DIAL

What is DIAL?

Design

Status

Development plans

GRID requirements

- Results and tasks
- Applications
- Schedulers
- Datasets
- Exchange format



Goals of DIAL

1. Demonstrate the feasibility of interactive analysis of large datasets
 - Large means too big for interactive analysis on a single CPU
2. Set requirements for GRID services
 - Datasets, schedulers, jobs, resource discovery, authentication, allocation, ...
3. Provide ATLAS with analysis tool
 - For current and upcoming data challenges



What is DIAL?

Distributed

- Data and processing

Interactive

- Prompt response (seconds rather than hours)

Analysis of

- Fill histograms, select events, ...

Large datasets

- Any event data (not just ntuples or tag)



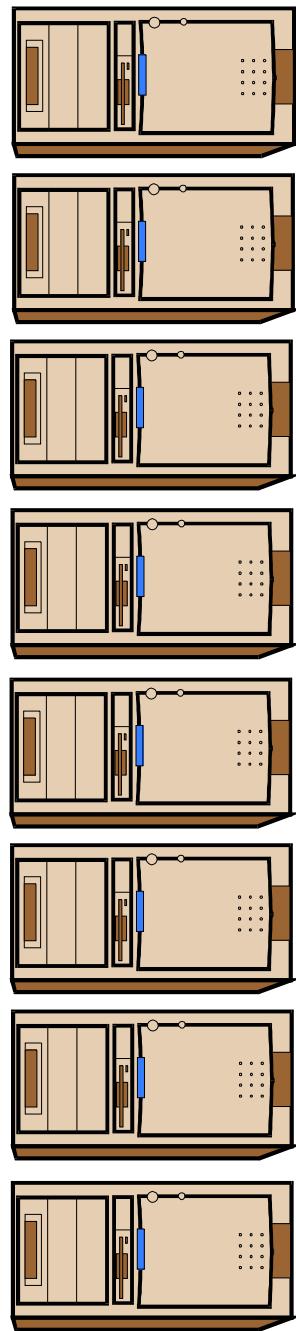
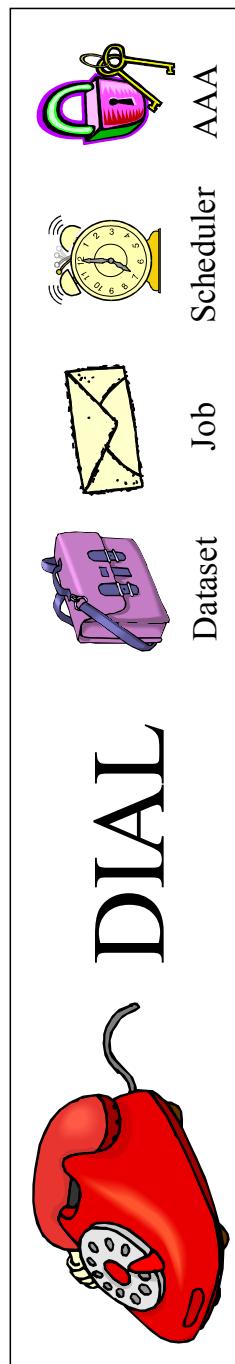
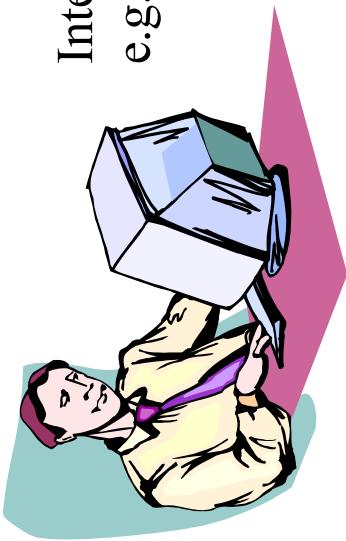
What is DIAL? (cont)

DIAL provides a connection between

- Interactive analysis framework
 - Fitting, presentation graphics, ...
 - E.g. ROOT
 - and Data processing application
 - E.g. athena for ATLAS
 - Natural for the data of interest
- DIAL distributes processing
- Among sites, farms, nodes
 - To provide user with desired response time



Interactive analysis
e.g. ROOT, JAS, ...



Distributed processing running data-specific application



David Adams
BROOKHAVEN
NATIONAL LABORATORY
PPDG

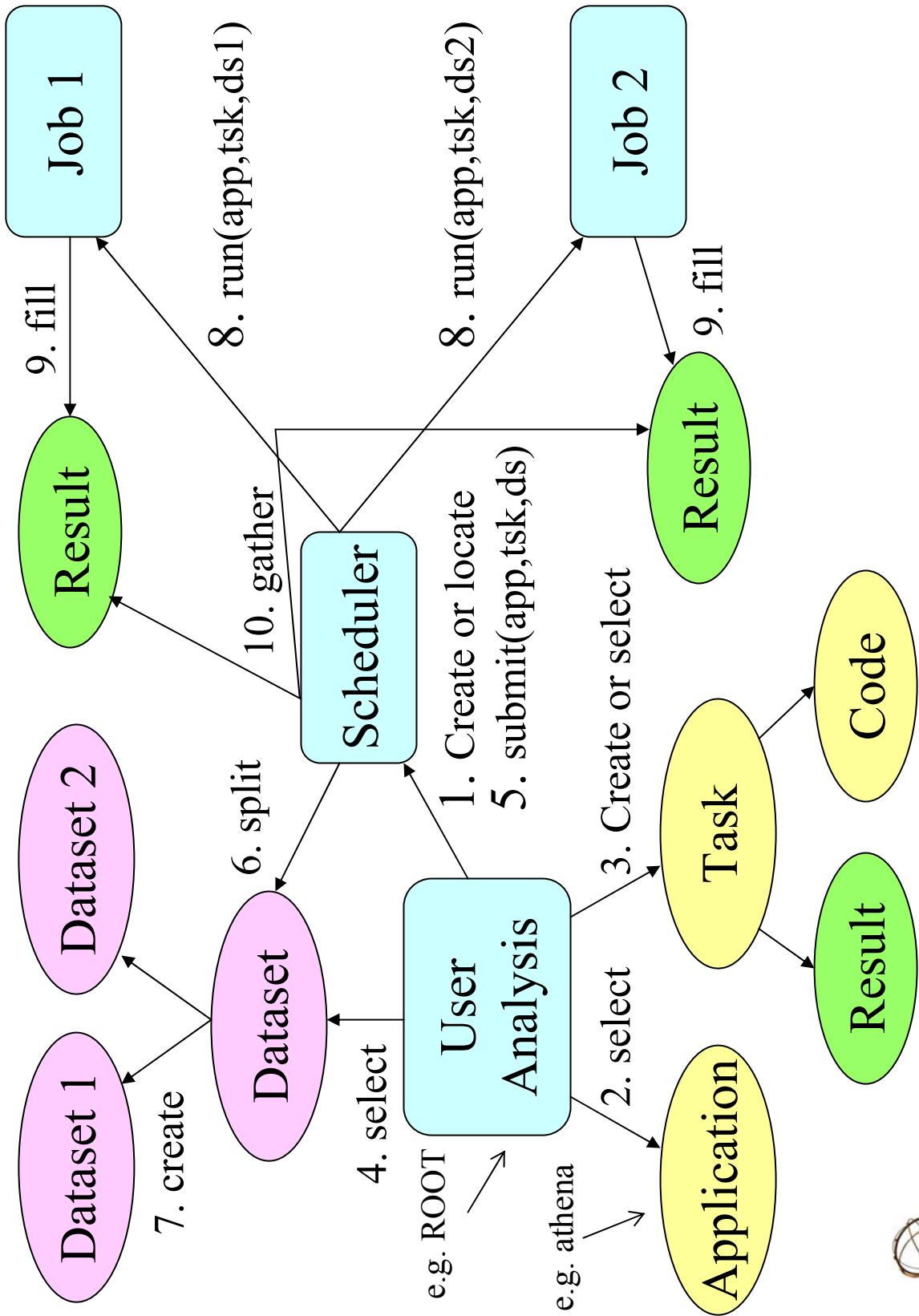
ATLAS SW – Grid session May 15, 2003

Design

Components of DIAL include

- Dataset describing the data of interest
 - Organized into events
- Application
 - Specification of the executable which loops over events and provides access to the data
- Task
 - Result to fill for each event
 - Code process each event
- Scheduler
 - Distributes processing and combines results





Results and Tasks

Result is filled during processing

- Examples

- Histogram
- Event list
- File

Task provided by user

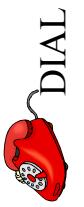
- Empty result plus
 - Code to fill the result
- Language and interface depend on application
 - May need to be compiled



Applications

Current application specification is

- Name
 - E.g. athena
 - Label – Not necessarily the executable file name
- Version
 - E.g. 6.1.3
- List of optional shared libraries
 - E.g. libRawData, libInnerDetectorReco



Applications (cont)

Each DIAL compute node provides an application description database

- File-based
 - Location specified by environmental variable
- Indexed by application name and version
- Application description includes
 - Location of executable
 - Run time environment (shared lib path, ...)
 - Command to compile task code
- This interface defined by ChildScheduler
 - Different scheduler could change conventions



Schedulers

A DIAL scheduler provides means to

- Submit a job
- Terminate a job
- Monitor a job
 - Status
 - Events processed
 - Partial results
- Verify availability of an application
- Install and verify the presence of a task for a given application



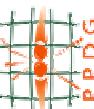
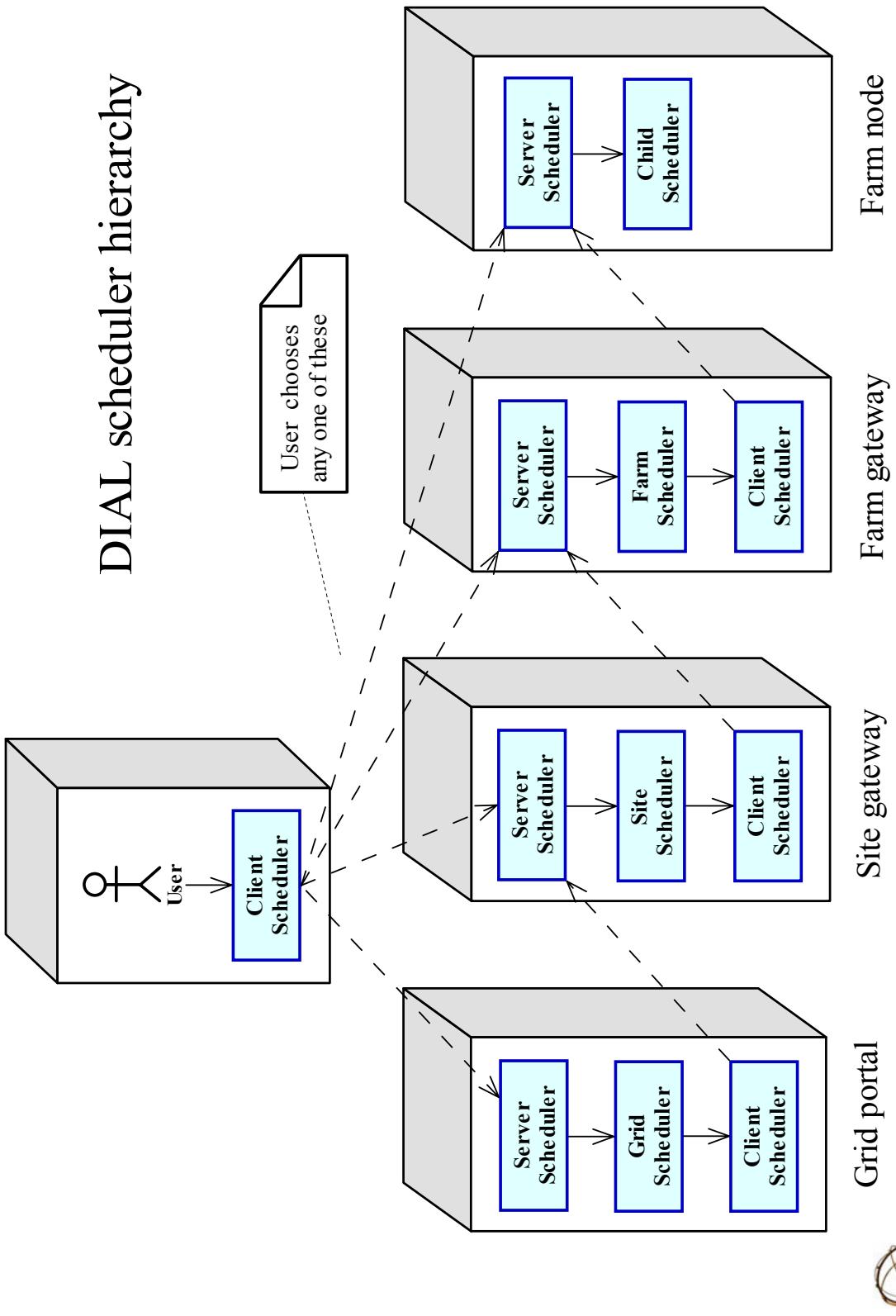
Schedulers (cont)

Schedulers form a hierarchy

- Corresponding to that of compute nodes
 - Grid, site, farm, node
- Each scheduler splits job into sub-jobs and distributes these over lower-level schedulers
- Lowest level ChildScheduler starts processes to carry out the sub-jobs
- Scheduler concatenates results for its sub-jobs
- User may enter the hierarchy at any level
- Client-server communication



DIAL scheduler hierarchy



David Adams
BROOKHAVEN
NATIONAL LABORATORY



Datasets

Datasets specify event data to be processed

Datasets provide the following

- List of event identifiers
- Content
 - E.g. raw data, refit tracks, cone=0.3 jets, ...
- Data location
 - List of logical files where data can be found
 - Mapping from event ID and content to file
 - Means to fetch data iterating over events
 - > E.g. ATLAS event collection



Exchange format

DIAL components are exchanged

- Between
 - User and scheduler
 - Scheduler and scheduler
 - Scheduler and application executable
- Components have an XML representation
- Exchange mechanism can be
 - C++ objects
 - SOAP
 - Files
- Mechanism defined by scheduler



Status

All DIAL components in place

- <http://www.usatlas.bnl.gov/~dladams/dial>
 - Release 0.20 made in March
 - But scheduler is very simple
 - Only local ChildScheduler is implemented
 - Grid, site, farm and client-server schedulers not yet implemented

More details in CHEP paper at

- http://www.usatlas.bnl.gov/~dladams/dial/talks/030325_dial.ppt



David Adams
BROOKHAVEN
NATIONAL LABORATORY

Status (cont)

Dataset implemented as a separate system

- <http://www.usatlas.bnl.gov/~dladams/dataset>
- Implementations:
 - ATLAS AthenaRoot file
 - > Exists
 - > Holds Monte Carlo generator information
 - ATLAS combined ntuple hbook file
 - > Under development
 - Athena-Pool files
 - > when they are available
- ATLAS datasets were described in a talk at the DB session yesterday



Status (cont)

DIAL and dataset classes imported to ROOT

- ROOT can be used as interactive user interface
 - All DIAL and dataset classes and methods available at command prompt
 - DIAL and dataset libraries must be loaded
- Import done with ACLiC
- Only preliminary testing done
- Need to add result for TH1 and any other classes of interest



Status (cont)

No application integrated to process jobs

- Except test program dialproc can be used to count events
- Plans for ATLAS:
 - Dialpaw to run paw to process combined ntuple
 - > Under development
 - Athena to process Athena-Pool event data files
 - > When Athena-Pool is available later this year
 - Perhaps a ROOT backend to process ntuples
 - > Or is this better handled with PROOF?
 - > Or use PROOF to implement a farm scheduler?



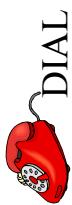
Status (cont)

Interface for logical files was added recently

- Includes abstract interface for a file catalog
 - Local directory has been implemented
 - Plan to add AFS catalog
 - > Good enough for immediate ATLAS needs
 - Eventually add Magda and/or RLS



David Adams
BROOKHAVEN
NATIONAL LABORATORY
PPD G



DIAL ATLAS SW – Grid session May 15, 2003 21

Status (cont)

Result interface is modified

- No longer a collection of products
- Instead concrete results directly implement a new Result interface
 - Implemented now: Counter and Event ID list
 - Plan to add result that is a collection of results
 - Working on result consisting of an hbook file
 - containing histograms
- For event data, plan to add ROOT histograms
 - or maybe ROOT file



Development plans

Highlighted items in

- red required for useful ATLAS tool and
- green to use it to analyze Athena-Pool data

Schedulers

- Client-server schedulers
- Farm scheduler
 - Allows large-scale test
- Site and grid schedulers
 - GRID integration
 - Interact with dataset, file and replica catalogs
 - Authentication and authorization



Development plans (cont)

Datasets

- Hbook combined ntuple
 - in development
- Interface to ATLAS POOL event collections
 - expected in summer
- ROOT ntuples ??

Applications

- PAW (with C++ wrapper)
- Athena for ATLAS
- ROOT ??



Development plans (cont)

Analysis environment

- ROOT implementation needs testing
- Import classes into LCG/SEAL? (Python)
- JAS? (java binding?)
- Athena-ASK??
- Ganga??



David Adams
BROOKHAVEN
NATIONAL LABORATORY
PPDG



ATLAS SW – Grid session

May 15, 2003 25

GRID requirements

Identify components and services that can be shared with

- Other distributed interactive analysis projects
 - PROOF, JAS
- Distributed batch projects
 - Production (AtCom, GRAT, Chimera)
 - Analysis (GANGA)
- Non-HEP event-oriented problems
 - Data organized into a collection of “events” that are each processed in the same way



GRID requirements (cont)

Candidates for shared components include

- Dataset
 - Event ID list
 - Content
 - File mapping
 - Splitting
- Application
 - Specification
 - Installation
- Task
 - Transformation = Application + Task



GRID requirements (cont)

Shared components (cont)

- Job
 - Specification (application, task, dataset)
 - Response time
 - Hierarchy (split into sub-jobs)
 > DAG?
- Scheduler
 - Accepts job submission
 - Splits, submits and monitors sub-jobs
 - Gathers and concatenates results
 - Returns status including results and partial results



GRID requirements (cont)

Shared components (cont)

- Logical files
 - Catalogs
 - Replication
- Authentication and authorization
- Resource location and allocation
 - Data, processing and matching



David Adams
BROOKHAVEN
NATIONAL LABORATORY PPDG



DIAL ATLAS SW – Grid session

May 15, 2003 29

GRID requirements (cont)

Important aspect is *latency*

- Interactive system provides means for user to specify maximum acceptable response time
- All actions must take place within this time
 - Locate data and resources
 - Splitting and matchmaking
 - Job submission
 - Gathering of results
- Longer latency for first pass over a dataset
 - Record state for later passes
 - Still must be able to adjust to changing conditions



GRID requirements (cont)

Avoid sharp division between interactive and batch resources

- Share implies more available resources for both
- Interactive use varies significantly
 - Time of day
 - Time to the next conference
 - Discovery of interesting events
- Interactive request must be able to preempt long-running batch jobs
 - But allocation determined by sites, experiments, ...

